Probabilistic Faster R-CNN with Stochastic Region Proposing: Towards Object Detection and Recognition in Remote Sensing Imagery

Dewei Yi, Jinya Su, and Wen-Hua Chen

Abstract

Object detection is one of the most important tasks involved in intelligent agriculture systems, especially in pest detection. This paper focuses on a most devastated agricultural disaster: grasshopper plagues. Grasshopper detection and monitoring is of paramount importance in preventing grasshopper plagues. This paper proposes a probabilistic faster R-CNN algorithm with stochastic region proposing, where a probabilistic region proposal network, an image classification network, and an object detection network are integrated to detect and locate grasshoppers. More specifically, in the proposed framework, the probabilistic region proposal network considers attributes (e.g. size, shape) of region proposals and the image classification network identifies the existence of grasshoppers while the object detection network scores recognition confidence for a region proposal. By integrating these three networks, the uncertainty can be passed from end to end, and the final confidence is obtained for each region proposal can be explicitly quantified. To enhance algorithm robustness, a stochastic region proposing algorithm is developed to screen region proposals rather than using a predetermined threshold. The proposed algorithm is validated by recently collected grasshopper datasets. The experimental results demonstrate that the proposed algorithm not only outperforms competing algorithms in terms of average precision (0.91), average missed rate (0.36), and maximum F_1-score (0.9263), but also reduces the false positive rate of recognising the existence of grasshoppers in an open field.

Index Terms

Object detection; Image recognition; Gaussian mixture models; Region proposal network;

I. Introduction

Agriculture monitoring systems can play a vital role in preventing pest outbreaks [1]. Pests bring severe threats to quality and storage of agricultural products, agricultural economy and food

security [2–4]; for example, the recent outbreak of desert locust in Africa. To prevent pest plagues and enable effective treatments, real-time information of pests (e.g. types of species, amount, and distribution) is essential. Currently, pests are usually detected by human beings. Manual monitoring, however, is labour-intensive and expensive for large farms or grasslands [4]. As suggested by [5–7], computer vision techniques could provide promising solutions for detecting pests to achieve pest control. In [5], a support vector machine (SVM) trained by edge features (histograms of oriented gradients) can efficiently identify aphids achieving a mean identification rate of 86.81% and an error rate of 8.91%. For precisely identifying thrips, the region indexes (e.g. the ratio of major diameter to minor diameter) and colour indexes (e.g. RGB channels) are extracted and then fed into an SVM classifier [6]. The experimental results demonstrate that the classification error can be less than 2.25%. Compared to conventional image processing techniques which depend on hand-crafted features, deep learning methods can automatically learn semantic and deeper features to enhance the performance significantly [8]. In [3], a deep residual network is trained for pest identification and its accuracy can reach 98.67% for classifying 10 classes. In addition, an improved pyramidal stacked de-noising auto-encoder is proposed in [7], where the accuracy for moth species identification can achieve 96.9% with pose estimation.

However, most studies on pest identification focus on image classification, where an image is only classified as a pest or not. Only classifying pests is insufficient for achieving effective pest control, where the number and locations of pests are also required [9, 10]. This type of problem could be regarded as a general problem in object detection where not only pests are detected in an image but also their number, location and distribution could be provided. In recent years, object detection and recognition have attracted extensive attention due to its significance in image understanding [11, 12]. In comparison to traditional machine learning methods (e.g. k-nearest neighbours [13], CART [14], random forest [15], SVM [16], dynamic Bayesian networks [17]), deep Convolutional Neural Networks (CNNs) can achieve better performance in many applications, such as traffic lane detection [18], salient object detection [19, 20], scene recognition [21], and pedestrian detection [22]. This is because deep CNNs are powerful in feature extraction and representation [23]. Since accurate pest detection also extensively relies on feature extraction and representation, deep CNNs based methods are used for extracting deep features of pests so as to realise precise pest detection. However, a pure deep Convolutional learning approach cannot capture model uncertainty [24–27], besides, convolutional networks may overfit quickly [24]. A pure deep convolutional learning approach (e.g. faster R-CNN) proposes a region proposal with

a pre-defined threshold of objectness so the uncertainty of region proposing and the information of image level is neglected. In addition, a pure deep convolutional neural network uses a fixed threshold to assess final score for generating results after non-maximum suppression. Such a manner overlooks to consider the uncertainty of region proposing. As showing in experimental results of this paper, a pure deep convolutional neural network does not perform very well in object detection.

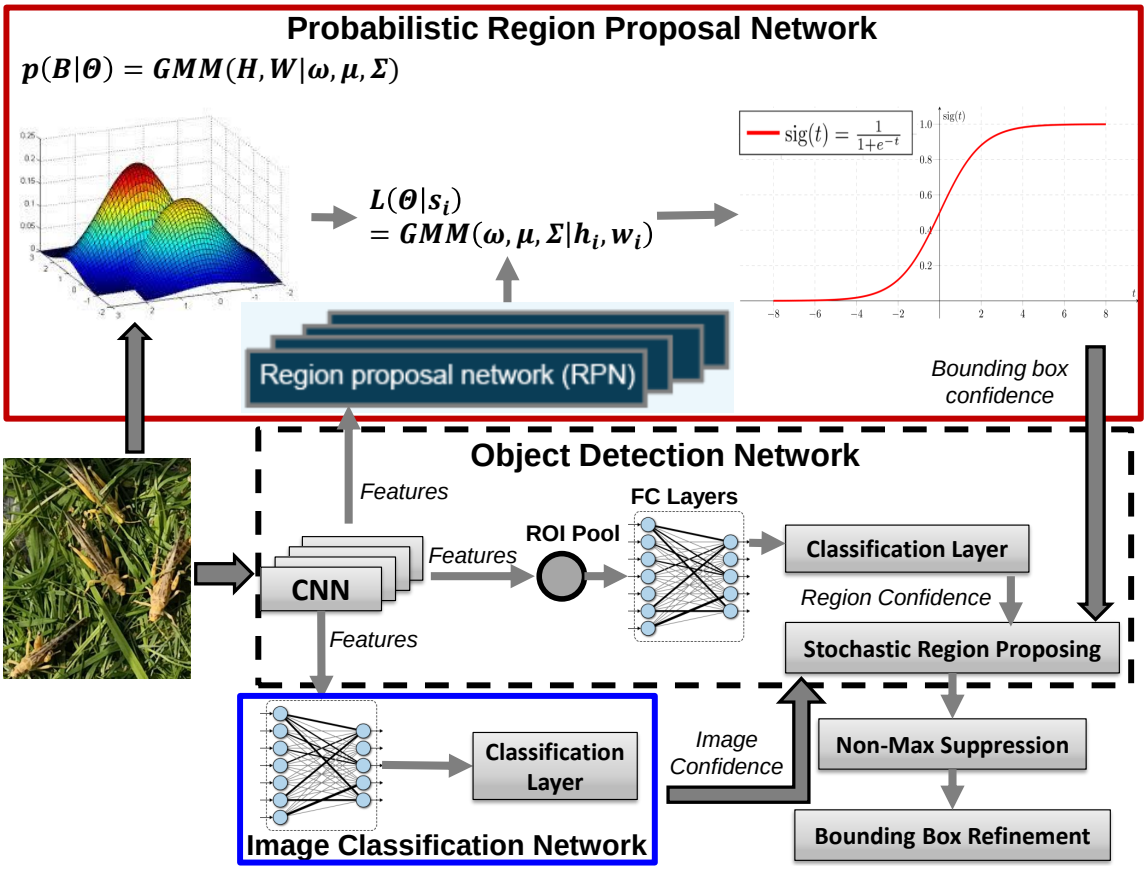


Fig. 1. The framework of probabilistic faster R-CNN with stochastic region proposing, where red solid rectangle represents the probabilistic region proposal network module, blue solid rectangle represents image classification network module, and black dash rectangle represents the object detection network module.

In order to capture and pass the uncertainty in entire object detection pipeline so as to further improve the performance, this paper proposes a probabilistic faster R-CNN with stochastic region proposing algorithm to detect and locate objects accurately. Our proposed algorithm mainly consists of three modules: probabilistic region proposal network, image classification network, and object detection network. The probabilistic region proposal network is used to

score goodness and objectness of bounding boxes; the image classification network is to enhance the classification accuracy at image level (object recognition) and reduce the false positive rate at object level (object detection), where goodness is to represent how well a bounding box is to capture a object and objectness is to identify whether a object exists within the bounding box; the object detection network is to obtain object level confidence score. In this study, we focus on detecting grasshoppers in preventing grasshopper plagues [28]. However, the proposed method could be transferred to other remote sensing applications such as weed detection [29], crop yield estimation [30], animal/wildlife monitoring (e.g. sheep/cow detection and counting) [31], forest management [32], and other pest detection applications (e.g. aphids or moths detection). Recently collected grasshopper datasets are adopted to evaluate the developed algorithms. Several baseline algorithms are also tested for comparison, which include aggregate channel features (ACF) [33], region-based Convolutional neural network (R-CNN) and its variations [34], fast R-CNN and its variations [35], faster R-CNN and its variations [36]. Comparative results demonstrate that the proposed algorithm substantially improves the performance at both image level and object level. This is because the proposed algorithm can pass the uncertainty from end to end, where the existence of object, the goodness of a region proposal, and the objectness of the proposed region proposal are represented in a probabilistic way. The uncertainty of object existence in image classification network is passed to region proposal network, which identifies the goodness of the region proposal. Then, the uncertainty of object existence and the goodness of region proposals is combined with the uncertainty of the objectness of the pixels within the bounding box for generating the final confidence score of the given object candidates. More precisely, the contributions of this paper are summarised as follows.

(i) A probabilistic region proposal network is proposed, which determines the confidence of a region proposal by considering both attributes (e.g. size and shape) and recognition score. Consequently, the uncertainty of a region proposal (goodness) can be measured in a probabilistic way and such uncertainty can be passed to the rest parts of object detection pipeline.

(ii) An image classification network is introduced into the proposed algorithm to reduce the errors in object recognition so that the errors of object detection can also be decreased;

(iii) The proposed algorithm integrates probabilistic region proposal network, image classification network, and object detection network together, which models and captures the

uncertainty of the entire object detection pipeline. Therefore, the final confidences are obtained for all possible region proposals to improve the performance of object detection.

(iv) To enhance the robustness, a stochastic region proposing algorithm is developed, which proposes bounding boxes in a stochastic way, rather than using a predetermined threshold.

The reminder of this paper is organised as follows. First, the overall framework of the proposed algorithm is illustrated in Section II, where the key components of the proposed algorithm are also introduced. Section III presents the probabilistic region proposal network while the image classification network is detailed in Section IV.Section V describes the object detection network and the stochastic region proposing algorithm, followed by experimental results discussion in Section VII. Section VIII concludes this paper along with future work.

II. Probabilistic Faster R-CNN with Stochastic Region Proposing

It is difficult to provide well-enough performance for locating objects and counting the number of objects accuracy by using existing methods [33–36] due to the failure of managing the uncertainty of the entire object detection pipeline. To tackle these issues and achieve satisfying object detection and recognition performance in remote sensing applications, we propose a probabilistic faster R-CNN with stochastic region proposing. This algorithm is composed of three modules. The first module is probabilistic region proposal network for generating confidences of region proposals. The second module is an image classification network which identifies the existence of objects within an image. The third module is to train a fast R-CNN detector using transfer learning with fine-tuning. Moreover, the duplicate region proposals are removed by non-maximum suppression [37]. The overall framework of the proposed algorithm is presented in Fig. 1. More specific explanations of the three modules and their responsibilities are provided as follows.

In the first module, the proposed probabilistic region proposal network is utilised to generate confidence scores of region proposals. The probabilistic region proposal network not only considers objectness scores of the box-classification layer, but also takes the size and shape of regions into account. Such a design improves the accuracy of recognising an object from the background.

In the second module, an image classification network is trained to identify the existence of objects at the image level. Since it is hard to generate sufficient data for learning from scratch, the image classification network is trained by transfer learning with fine-tuning. The

image classification network provides confidence ranging from 0 to 1. The confidence is used to identify the probability that the image includes at least one object. This image level confidence can affect the confidence of region proposal at the object level.

In the third module, an object detection network is adopted with stochastic region proposing. Firstly, the object detection network is to classify potential regions into objects or background based on the corresponding confidence scores of each class. These confidence scores combine with the uncertainty of the probabilistic region proposal network and the image classification network to generate the final confidence score for each potential region. Secondly, the proposed stochastic region proposing algorithm suggests regions stochastically according to the final confidence score of each region. Such a design can efficiently remove incorrect regions and enhance the robustness. Then, the final confidence scores of retained regions are used in non-maximum suppression to eliminate the duplicate detections for the same object.

Different from faster R-CNN which proposes a region proposal with a pre-defined threshold of objectness so the uncertainty of region proposing and the information of image level is neglected, our proposed probabilistic faster R-CNN passes uncertainty from end to end so that the final confidence of a region proposal $p(G_{ij}|b_{ij}, r_{ij}, O_i, I_i)$ integrates the uncertainty of the probabilistic region proposal network, the image classification network, and the object detection network for improving the object detection performance, where G_{ij} is the priori of interested object (e.g. pest, sheep, tree) for j-th region proposal in i-th image. r_{ij} and b_{ij} represent j-th region proposal and j-th bounding box in i-th image, respectively. O_i and I_i denote the interested object existence of object detection network and image classification network for i-th image. Thereby, Equation (1) can be derived by using chain rule.

$$p(G_{ij}|r_{ij}, b_{ij}, O_i, I_i) = \frac{p(r_{ij}, b_{ij}|G_{ij}, O_i, I_i)p(G_{ij}|O_i, I_i)}{p(r_{ij}|b_{ij}, O_i, I_i)p(b_{ij}|O_i, I_i)} \tag{1}$$

Because the confidence of region proposal r_{ij} and bounding box b_{ij} in i-th image is independent of interested object existence (O_i and I_i). The right hand of Equation (1) can be simplified to

$$\begin{aligned} &\frac{p(r_{ij}, b_{ij}|G_{ij}, O_i, I_i)p(G_{ij}|O_i, I_i)}{p(r_{ij}|b_{ij}, O_i, I_i)p(b_{ij}|O_i, I_i)} \\ &= \frac{p(r_{ij}, b_{ij}|G_{ij})p(G_{ij}|O_i, I_i)}{p(r_{ij}|b_{ij})p(b_{ij})} \end{aligned} \tag{2}$$

Moreover, the confidence of region proposal r_{ij} is conditionally independent of the shape of the bounding box b_{ij} given G_{ij}. Hence, Equation (2) reduces to

$$\frac{p(r_{ij}|G_{ij})p(b_{ij}|G_{ij})p(G_{ij}|O_i,I_i)}{p(r_{ij})p(b_{ij})} \quad . \tag{3}$$

According to Bayesian theorem [38], Equation (3) can be written as

$$\frac{p(r_{ij}|G_{ij})p(b_{ij}|G_{ij})p(G_{ij}|O_i,I_i)}{p(r_{ij})p(b_{ij})} = \frac{p(G_{ij}|r_{ij})p(G_{ij}|b_{ij})}{p^2(G_{ij})}\frac{p(G_{ij},O_i,I_i)}{p(O_i,I_i)} \tag{4}$$

where $p(G_{ij}|r_{ij})$ is the confidence that j-th region proposal in i-th image is a interested object, which comes from object detection network. $p(G_{ij}|b_{ij})$ is the confidence that region proposal r_{ij} is a correct bounding box, which is determined by the probabilistic region proposal network.

In order to fully use the confidence of image classification $p(G_{ij}|I_i)$, Equation (4) is expressed as

$$\frac{p(G_{ij}|r_{ij})p(G_{ij}|b_{ij})}{p^2(G_{ij})}\frac{p(G_{ij},O_i,I_i)}{p(O_i,I_i)} = \frac{1}{p^2(G_{ij})}p(G_{ij}|r_{ij})p(G_{ij}|b_{ij})\frac{p(O_i|G_{ij},I_i)}{p(O_i|I_i)}p(G_{ij}|I_i) \tag{5}$$

where $p(G_{ij}|I_i)$ is the confidence having interested object(s) within i-th image, which comes from image classification network. O_i is the determination of interested object existence from the object detection network, where O_i^+ and O_i^- representing having or not having interested object(s) within i-th image. G_{ij} is the truth of interested object existence. $p(O_i|G_{ij},I_i)$ is the probability distribution of the interested object existence recognised by object detection network when given the result of image level classification I_i. Moreover, $p(G_{ij}|I_i)$ is the probability distribution of the interested objects when given the result of image level classification I_i.

With regard to Equation (1)-(5), the final confidence of a region proposal can be defined as

$$\begin{aligned} &p(G_{ij}|r_{ij},b_{ij},O_i,I_i) = \\ &\frac{p(O_i|G_{ij},I_i)}{p^2(G_{ij})}p(G_{ij}|r_{ij})p(G_{ij}|b_{ij})\frac{p(G_{ij}|I_i)}{p(O_i|I_i)}, \end{aligned} \tag{6}$$

where $p(G_{ij})$ is the priori probability of interested object detection. $p(G_i|I_i)$ can be obtained by the statistics of results, which comes from the training set. However, it is difficult to directly obtain the values of $p(G_{ij})$ and $p(O_i|G_i,I_i)$. To tackle this issue, $p(O_i|G_i,I_i)/p^2(G_{ij})$ is regarded as a hyperparameter.

III. Probabilistic Region Proposal Network

Probabilistic region proposal network assesses confidences of potential region proposals by considering both objectness scores and attributes of region proposals (e.g. the size and shape

of the bounding boxes). Inspired by [39], Gaussian mixture models are introduced to assess the attributes of region proposals. Probabilistic region proposal network includes three key components: region proposal network, Gaussian mixture models, and probabilistic inference. The mechanism of the probabilistic region proposal network is displayed in Fig. 2

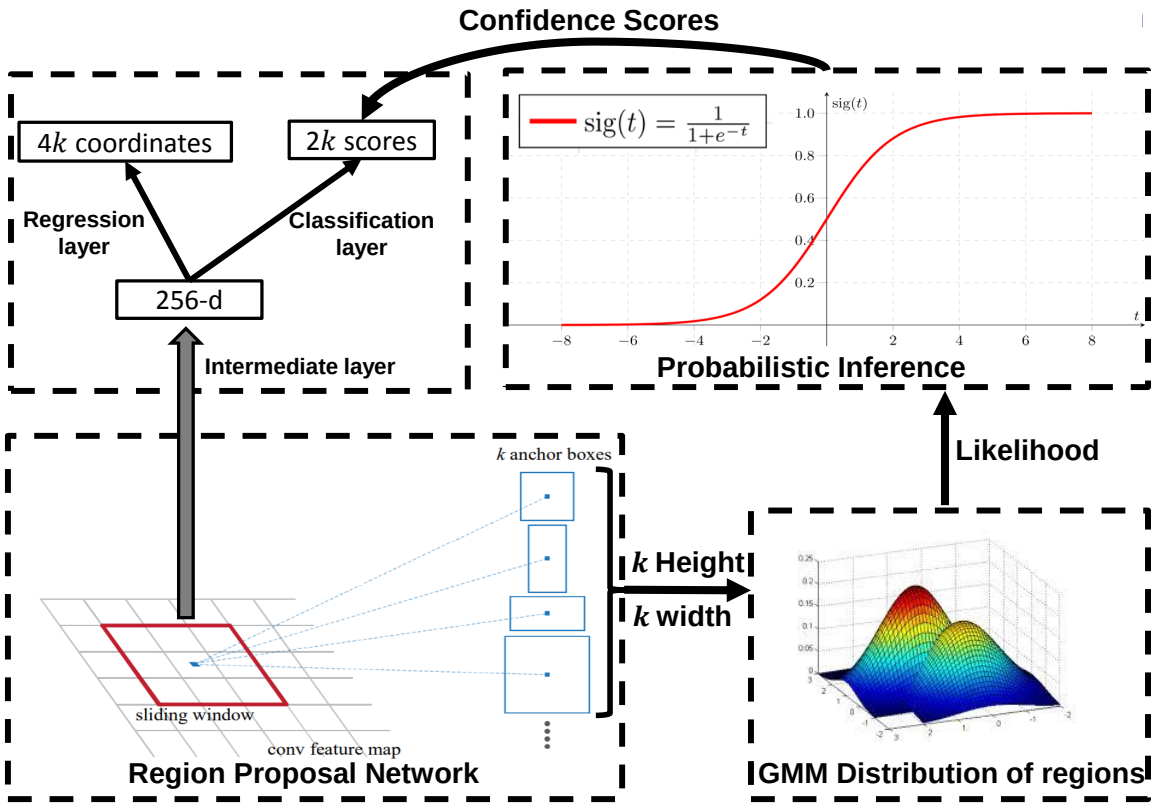


Fig. 2. Mechanism of the proposed Probabilistic Region Proposal Network.

A. *Region Proposal Network*

The region proposal network proposes a set of rectangular object proposals for an inputted image. This process is carried out by a fully convolutional network and a small network. The small network slides over the convolutional feature map with a fixed-size window [35]. The features of each sliding window are fed into a box-regression layer and a box-classification layer, where the box-regression layer generates a rectangular object proposal while the box-classification layer determines with or without an object. For more details, please refer to [36].

B. Gaussian Mixture Models

The inputs of Gaussian mixture models (GMM) should be defined before constructing GMM. According to manually labelled bounding boxes, we can determine the goodness of a bounding box based on the height and width of the manually labelled bounding boxes. The height and width are inputs of GMM. After given the height and width of bounding boxes, we can obtain the probability density function (PDF) via GMM. The GMM is a weighted sum of M component Gaussian densities as follows:

$$p(B|\Theta) = \sum_{i=1}^{M} w_i g(B|\mu_i, \Sigma_i) = \sum_{i=1}^{M} w_i g(H, W|\mu_i, \Sigma) \tag{7}$$

where B is a D-dimensional matrix. Each row of B is a data sample. Here, X is a two-dimensional matrix. These two dimensions represent the height (H) and width (W) of bounding boxes ($B = \{H, W\}$). M is the number of Gaussian components, w_i is the mixture weight for i-th Gaussian component, and $g(B|\mu_i, \Sigma_i)$ is the density function of i-th Gaussian component. The density of each component is a two-variate Gaussian function of the following form:

$$p(B|\mu_i, \Sigma_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(B-\mu_i)^T \Sigma_i^{-1}(B-\mu_i)} \tag{8}$$

where μ_i is the mean vector. In this paper, the mean vector includes the means of height and width for bounding boxes. Σ_i is a covariance matrix of height and width. Moreover, the mixture weight w_i satisfies the following constraint

$$\sum_{i=1}^{M} w_i = 1 \tag{9}$$

Consequently, the complete GMM is parametrised by the mean vectors, covariance matrices, and mixture weights of all component densities, which are collectively represented by the notation.

$$\Theta = w_i, \mu_i, \Sigma_i, i = 1, \ldots, M. \tag{10}$$

There are two issues involved in establishing GMM: determining the optimal component number and deriving best matching between a generated model and its training data through parameter optimisation. In this paper, the optimal component number is determined by Akaike's Information Criterion (AIC) [40] and Bayesian Information Criterion (BIC) [41]. For more details, please refer to Section VII-A. Meanwhile, the parameters Θ in Equation (10) are optimised by expectation-maximisation (EM) algorithm [39] and the initialisation of parameters is implemented by k-means ++ clustering algorithm [42]. Subsequently, the likelihood of GMM can be maximised. The main steps of optimising GMM are summarised in *Algorithm 1*.

Algorithm 1 Gaussian Mixture Models with Optimisation

Require: Given component number k, $k \in \{1, 2, \ldots, N\}$, initial mixture parameters $\Theta^{(0)}$, and convergence threshold ϵ;

Initialisation: $\Theta^{(t)} = \Theta^{(0)}$
for $k = 1 : N$ **do**
  for $t = 1 : +\infty$ **do**
    E-step: Compute the posteriori probability
    $\hat{\gamma}_{ik} = p(z_i = k|b_i, \Theta^{(t)}) = \frac{w_i g(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{M} w_j g(x_i|\mu_j, \Sigma_j)}$
    M-step: Maximisation of the likelihood
    $\hat{w}_k = \frac{1}{n}\sum_{i=1}^{n} \hat{\gamma}_{ik}$, $\quad \hat{\mu}_k = \frac{\sum_{i=1}^{n} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{n} \hat{\gamma}_{ik}}$
    $\hat{\Sigma}_k = \frac{\sum_{i=1}^{n} \hat{\gamma}_{ik}(x_i - \hat{\mu}_k)^T (x_i - \hat{\mu}_k)}{\sum_{i=1}^{n} \hat{\gamma}_{ik}}$
    $\Theta^{(t)} = \{\hat{w}_k, \hat{\mu}_k, \hat{\Sigma}_k\}$
    if $L(B|\Theta^{(t)}) - (B|\Theta^{(t-1)}) \leq \epsilon$ **then**
      Set mixture parameters $\Theta = \Theta^{(t)}$
  Calculate AIC(k) and BIC(k) of GMM(k)
  if AIC(k) & BIC(k) $<$ AIC(k^*) & BIC(k^*) **then**
    Set optimal component number $k^* = k$

Ensure: obtained the optimal parameters for k^* and Θ

C. Probabilistic Inference

After obtaining GMM for establishing the relationship between the attributes of region proposals and the likelihood of a correct bounding box in Section III-B, the likelihood of a given region proposal can be computed by

$$L(\Theta|b_{ij}) = p(\Theta|b_{ij}) = \sum_{z=1}^{M} w_z g(\mu_z, \Sigma_z|h_{ij}, w_{ij}) \tag{11}$$

where b_{ij} is a new region proposal. To improve robustness of the probabilistic inference, instead of determining region proposals using a specific threshold of likelihood, Sigmoid function is introduced to normalise likelihood with in [0, 1] which can return the probability of the proposed region being a correct bounding box. In this paper, Sigmoid function is defined as

$$Sig(t) = \frac{1}{1 + e^{-t}} = \frac{1}{1 + e^{-\alpha(L_{ij} - L_{min})}} \tag{12}$$

where L_{ij} is the likelihood for a given a new bounding box b_{ij} and L_{min} is the minimum likelihood as a positive bounding box in the training set. Moreover, α is a factor to tune the sensitivity. Following [43], it is set as 1 for achieving harmonic results.

It follows from Equations (11) and (12) that the confidence $p(G_{ij}|b_{ij})$ of the region proposal r_{ij} being a correct bounding box can be given by

$$p(G_{ij}|b_{ij}) = \left[1 + exp\Big(\alpha L_{min} - \alpha \sum_{z=1}^{M} w_z g(\mu_z, \Sigma_z | h_{ij}, w_{ij})\Big)\right]^{-1} \tag{13}$$

IV. Image Classification Network

Different from faster R-CNN, the CNN in probabilistic faster R-CNN does not only share weights of convolutional layers with the region proposal network but also executes the image classification to determine the existence of grasshoppers in an image. The image classification is also achieved by a CNN, which shares the same weights of convolutional layers with the region proposal network and the object detection network. The image classification network returns the confidence whether grasshoppers exist in the image, which is represented by $p(G_{ij}|I_i)$ in Equation (6).

Due to such a design, classification results from the image level can affect the confidences of the region proposals at the object level. For instance, regions have a lower probability to be proposed when an image is classified to be a low confidence of having grasshoppers. O_i is the determination of grasshopper existence from the object detection network, where O_i^+ and O_i^- represent for positive and negative recognition results in i-th image respectively. The conditional probability of $p(O_i|I_i)$ can be calculated by Table I, where t is a specific threshold. In addition, hyperparameter $p(O_i|G_i, I_i)/p^2(G_{ij})$ is optimised by a common heuristic approach [39], where the search range and interval for optimising the hyperparameter are $[0.001, 10]$ and 0.001, respectively.

TABLE I
Summary of Conditional Probabilities

Truth/Image	Grasshopper	Non-Grasshopper
	$p(O_i\|I_i)$	
Grasshopper	$p(O_i^+\|I_i \geq t)$	$p(O_i^+\|I_i < t)$
Non-Grasshopper	$p(O_i^-\|I_i \geq t)$	$p(O_i^-\|I_i < t)$

V. Object Detection Network with Stochastic Region Proposing

In this paper, a fast R-CNN is used as the object detection network due to its excellent performance on computational efficiency [35, 36]. The confidence of the existing of a grasshopper in each region proposal is identified by the fast R-CNN, which is represented by $p(G_{ij}|r_{ij})$. The priori probability $p(G_{ij})$ and the conditional joint probability of $p(O_i|G_{ij}, I_i)$ are difficult to calculate directly. Hence, their combination $p(O_i|G_i, I_i)/p^2(G_{ij})$ is treated as a hyperparameter which is optimised during the training phase. As mentioned in Section III and Section IV, $p(G_{ij}|b_{ij})$ and $p(G_{ij}|I_i)$ can be obtained from the probabilistic region proposal network and the image classification network. Moreover, the method to calculate $p(O_i|I_i)$ is provided in Section IV. From the above, the final confidence of a region proposal $p(G_{ij}|r_{ij}, b_{ij}, O_i, I_i)$ can be computed by Equation (6).

Algorithm 2 Stochastic Region Proposing Algorithm

Require: obtained the region proposal r_{ij} and the image i, $i \in \{1, 2, \ldots, I\}$ and $j \in \{1, 2, \ldots, R\}$;

1: **for** $i = 1 : I$ **do**
2: **for** $j = 1 : R$ **do**
3: Set final confidence $p(G_{ij}|r_{ij}, b_{ij}, O_i, I_i)$ = p
4: Randomly generate a number n, $n \in [0, 1]$
5: **if** $n \leq p$ **then**
6: Set $r_s = \{r_{s-1}, r_{ij}\}$

Ensure: selected region proposals r_s

To improve the robustness and make full use of probabilistic information, a stochastic region proposing algorithm is developed to generate region proposals with its pseudocode in *Algorithm 2*. It removes incorrect region proposals and then non-maximum suppression is introduced to merge the nearby detections of the same grasshopper according to the final confidence scores of the retained region proposals.

VI. Implementation and Performance Metrics

Implementation and performance metrics are discussed in this section. All experiments are evaluated by a natural field grasshopper dataset, which is described in Section VI-A. To quantitatively evaluate the performance at both image level and object level, a number of widely used

metrics are adopted, including intersection-over-minimum (IoM) [9], miss rate, false positives per image ($FFPI$) [33], precision, recall, specificity, accuracy and F_β-score as defined in Section VI-B. ACF algorithm is implemented by a CPU (Core i7 at 2.50 GHz with 16 GB of RAM), which is a non-deep object detection algorithm. The remaining algorithms are deep learning object detection methods, which are implemented by a GPU (GeForce RTX 2080). In addition, the parameter settings of ACF is referred to [10], where the number of stages is 6 and negative samples factor is 4. The parameter settings of deep learning based methods are provided as follows: the *optimiser* is stochastic gradient descent with momentum (SGDM), *MaxEpochs* is 20, and *InitialLearnRate* is $1 \times e^{-3}$.

A. Grasshopper Detection and Recognition Dataset

1) Data Acquisition: RGB colour images were obtained from an outdoor test field at Loughborough University, Leicestershire, UK. The image dataset is collected by the Leica dual camera (2MP, CMOS) mounted at the right above each cell of the test field and under natural light conditions. The date acquisition of cage and open field datasets are conducted in the same way. The only difference is that grasshoppers are within a cage or an open field. Imaging in an outdoor environment under different weather condition is important for training a robust model that is applicable to a wide range of weather conditions. Images were captured at 60 cm distance right above the test field.

2) Dataset Construction: The collected images are randomly divided into two sets: the training set (60%) and validation set (40%) for both cage and open field datasets. For each set, we calculate the number of images with or without grasshoppers, the total number of grasshoppers, and the number of grasshoppers per image in corresponding sets. Table II provides specific statistics of the glass cage dataset and the open field dataset including the training, validation, and entire sets, respectively.

B. Evaluation Metrics

The performance of the grasshopper detection can be evaluated by correct detections (true positive), miss detections (false negative), and false positive. A miss detection is determined based on the manually labelled region. When the algorithm proposes a region (bounding box) and does not have a manually labelled region corresponding with it, this is called by a miss detection and the proposed region is recognised as a false negative. In this paper, we evaluate the

TABLE II

SUMMARY OF GRASSHOPPER DATASETS

Glass Cage Grasshopper Dataset					
Dataset	Total #	# images with grasshoppers	# images without grasshopper	# grasshoppers	Avg. # grasshoppers per image
Total	857	316	541	5010	5.84
Training	514	189	325	2970	5.78
Validation	343	127	216	2040	5.95
Open Field Grasshopper Dataset					
Dataset	Total #	# images with grasshoppers	# images without grasshopper	# grasshoppers	Avg. # grasshoppers per image
Total	3412	1712	1700	20544	6.02
Training	2047	1037	1010	12444	6.08
Validation	1365	675	690	2040	5.93

performance at both object level and image level. At the object level evaluation, we focus on the performance of detecting individual grasshoppers. At the image level evaluation, we focus on the performance of identifying whether grasshoppers are within the image or not. To determine a correct detection, we compute the intersection-over-minimum (IoM) between the detection and its corresponding manually labelled bounding box. The definition of IoM is given by

$$IoM(A_{dt}, A_{gt}) = \frac{area(A_{dt} \cap A_{gt})}{min(A_{dt}, A_{gt})} \tag{14}$$

where A_{dt} is the area of a detected bounding box and A_{gt} is the area of a ground truth bounding box. IoM is the intersection between the detected bounding box and ground truth bounding box over the minimum area of the detected bounding box and ground truth bounding box. We consider a correct detection if $IoM \geq 0.5$. Otherwise, the detection is identified as a false detection. In this paper, we use IoM rather than intersection-over-union (IoU). This is because IoU performs better when the ground truth rectangles are more nearly square and IoM performs better when the ground truth rectangles are more nearly tall or wide rectangles [9]. In the grasshopper detection, ground truth rectangles are always tall or wide rectangles rather than squares [9].

1) Object Level Evaluation: To evaluate the performance at object level, five threshold-dependent measures are used including miss rate, false positives per image ($FFPI$), precision, recall, and F_β score. The definitions of these five metrics are given as follows:

$$MissRate = \frac{d_m}{d_g}, \quad FFPI = \frac{f_p}{n} \tag{15}$$

where d_m is the number of miss detections of grasshoppers and d_g is the total number of grasshoppers. f_p is the number of false positives and n is the total number of images. In this case,

f_p represents the number of miss detected grasshoppers (The proposed regions are background, but detected as grasshoppers).

$$Precision = \frac{t_p}{t_p + f_p}, \quad Recall = \frac{t_p}{t_p + f_n} \tag{16}$$

where t_p is the number of true positives and f_n is the number of false negatives. In this case, t_p represents the number of correct detected grasshoppers and f_n represents false detected grasshoppers (The proposed regions are grasshoppers, but they are detected as background). Therefore, the sum of t_p and f_p is the total number of detected grasshoppers (Prediction). The sum of t_p and f_p is the total number of grasshoppers (Ground Truth).

Moreover, F_β is introduced which considers precision and recall simultaneously, where β is the parameter for adjusting the importance of precision and recall. When β exceeds one, it means that precision is more important. Otherwise, recall is more important. In this paper, precision and recall are both significant for detecting grasshoppers so we use F_1-score by setting $\beta = 1$ which applies the same weight to precision and recall [39]. The definition of F_β is provided as follows.

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{17}$$

In practice, we receive different precisions and recalls according to different thresholds. It results in different F_1-scores. Here, the maximum value of F_1-scores is used to assess the object level performance. In addition, Precision vs. recall curve measures the trade-off between precision and recall. The area under the Precision-Recall curve is called average precision, which summarises the weighted increase in precision with each change in recall of different thresholds. The definition of average precision (AP) is given by

$$AP = \sum_{i=1}^{n} (R_i - R_{i-1}) P_i \tag{18}$$

where P_i and R_i are the precision and recall at the i-th threshold and n is the total number of thresholds. Therefore, AP is a single number for indicating the object detection performance with varying thresholds.

Similar to miss rate and $FPPI$, miss rate vs. $FPPI$ curve measures the performance under certain tolerances specified by the number of false positives. Such curve is called average miss rate, which is computed by nine $FFPI$ rates evenly spaced in log-space in the range 10^{-2} to 10^0 [44].

2) Image Level Evaluation: To access the performance of image level, five metrics are selected including sensitivity, specificity, precision, and F_β score. Sensitivity is defined as the same as recall. However, at the image level, sensitivity is the number of correctly identified true grasshopper images over the total number of true grasshopper images. The specificity is defined as follows:

$$Specificity = \frac{t_n}{t_n + f_p} \tag{19}$$

where t_n are the true negatives. Here, t_n represents the number of correctly identified no grasshopper images. The sum of t_n and f_p is the total number images without grasshopper. At image level, precision is the number of correctly identified true grasshopper images over the total number of images which are identified with grasshopper(s) inside. The setting of F_β follows the suggestion in Section VI-B1, where $\beta = 1$. Because image level evaluation is formulated as an image classification problem, confusion matrices are also provided.

VII. Experimental Evaluation

In this section, experimental evaluation is conducted on the proposed algorithm for grasshopper detection. In the training phase, the component number of GMM is optimised by Akaike's information criterion and Bayesian information criterion, which is investigated in Section VII-A. To quantitatively evaluate the performance of image level and object level, different metrics are used including intersection-over-minimum (IoM) [9], miss rate, false positives per image ($FFPI$) [33], precision, recall, specificity, accuracy and F_β-score. The performance of the proposed algorithm is compared against other algorithms in Section VII-B.

A. Tuning Gaussian Mixture Models

It is usually challenging to tune the component number k^* for GMM for a specific application. In order to derive "optimal" k^*, Akaike's Information Criterion (AIC) [40] and Bayesian Information Criterion (BIC) [41] are both introduced which are effective measures for assessing the quality of mixture models. According to Akaike's theory and Bayesian theory, the most appropriate model has the smallest AIC value and BIC value. The definition of AIC and BIC are provided by Equation (20) and Equation (21) respectively.

$$\begin{aligned} AIC = n \times \log\Big(\det(\frac{1}{n}\sum_{i=1}^{n}\epsilon(t,\hat{\theta}_i)^T\epsilon(t,\hat{\theta})\Big) + 2n_p \\ +N \times (\log(2\pi)+1) \end{aligned} \tag{20}$$

$$BIC = n \times \log\Big(\det(\frac{1}{n}\sum_{i=1}^{n}\epsilon(t,\hat{\theta}_i)^T\epsilon(t,\hat{\theta})\Big) + n_p \times \log(n) + N \times (\log(2\pi)+1) \tag{21}$$

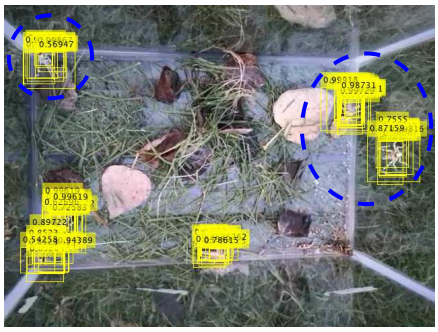
where n is the number of data samples. $\epsilon(t)$ is a 1-by-n_y vector of prediction errors, where n_y is the number of model outputs. θ_i is the estimated parameters for i-th data sample with parameter number of n_p. For a given number of components, the corresponding values of AIC and BIC can be computed. The values of AIC and BIC with different component numbers ranging from 2 to 7 are calculated in Table III. It can be seen that the "optimal" components number of GMM is 6 since both AIC and BIC reach the minimum values.

TABLE III
SUMMARY OF GRASSHOPPER DATASET

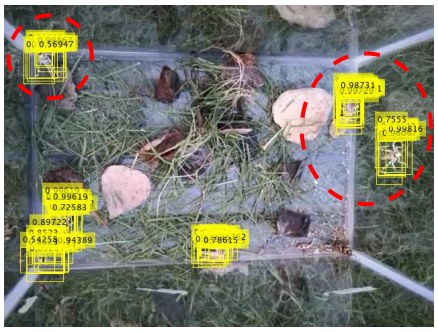
Component No.	AIC($\times 10^4$)	BIC($\times 10^4$)
2	4.5247	4.5301
3	4.4515	4.4599
4	4.4179	4.4293
5	4.3939	4.4083
6	**4.3241**	**4.3415**
7	4.3451	4.3655

B. Performance of Grasshopper Detection and Recognition

In this section, the performance is evaluated at region proposal level, image level, object level and their combination. Deep neural networks require a large amount of labelled data to train from scratch. However, collecting sufficient data and manually labelling these data might be infeasible in many cases like grasshopper detection and recognition. A promising alternative is transfer learning with fine-tuning. In this paper, transfer learning is used both in object detection and image classification and the parameters of transferred neural networks are fine-tuned. At object level, a faster R-CNN object detection network is trained. It is difficult to determine the number of layers to be transferred for grasshopper detection. Here, we attempt to make full use of the low-level features from a pre-trained deep neural network, where the features are generated from convolutional layers. Alexnet [45] is used as a pre-trained deep neural network for conducting transfer learning and fine-tuning. Subsequently, fully connected layer(s) are replaced and the new obtained network is trained with grasshopper data again for the sake of fine-tuning.



(a) Region proposal network (glass cage dataset)



(b) Probabilistic region proposal network (glass cage dataset)

(c) Region proposal network (open field dataset)

(d) Probabilistic region proposal network (open field dataset)

Fig. 3. Advantages of probabilistic region proposal network and image classification network (ICN). Blue dash circles in (a) show the results of region proposal network and red dash circles in (b) show the results of probabilistic region proposal network on glass cage dataset; Blue dash circles in (c) show the results of region proposal network and red dash circles in (d) show the results of probabilistic region proposal network on open filed dataset. We can see that some incorrect region proposals are removed with using ICN showing in red cycles compared to without using ICN showing in blue cycles.

We compared the results of transferring all fully connected (AFC) layers and only last fully connected (LFC) layer. More details are provided in Section. V.B.2. At image level, we conduct transfer learning and fine-tuning for the pre-trained network following the suggestions from [46].

1) Incorrect Region Proposals Removal: The proposed probabilistic region proposal network not only considers the category (grasshopper or background) of proposed regions, but also takes the goodness of the proposed bounding boxes into account. The latter is estimated by the likelihood of the GMM. Such a design helps remove incorrect proposals by stochastic region

proposing. As illustrated by the comparison between Fig. 3(a), Fig. 3(c) and Fig. 3(b) and Fig. 3(d), some incorrect region proposals are removed by considering the attributes of region proposals, which is highlighted by red dash circles compared to blue dash circles.

2) Object Level Performance: To evaluate the performance of the proposed algorithm at object level, several algorithms are implemented and compared in grasshopper detection and recognition, which includes ACF [33], R-CNN with transferring last or all fully connected layer(s) [34], faster R-CNN with transferring last or all fully connected layer(s) [36], our proposed faster R-CNN with bounding box constraints (BBC), and our proposed probabilistic faster R-CNN. Because this section focuses on object level performance evaluation, probabilistic faster R-CNN does not integrate with image classification network (ICN). The comparative results in terms of average precision (AP), average miss rate (AMR) and maximum F_1-score are summarised in Table IV and V.

TABLE IV
COMPARISON OF OBJECT LEVEL ON GLASS CAGE DATASET

Algorithm	Average Precision (AP)	Average Miss Rate (AMR)	Max F_1-score
ACF	0.39	0.85	0.4784
R-CNN + Transfer AFC	0.53	0.60	0.6725
R-CNN + Transfer LFC	0.50	0.58	0.6474
Faster R-CNN + Transfer AFC	0.52	0.78	0.5558
Faster R-CNN + Transfer LFC	0.77	0.58	0.7990
Ours (Faster R-CNN + Transfer LFC + BBC)	0.78	0.56	0.7979
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP without ICN)	**0.79**	**0.55**	**0.8059**

The following observations can be made from Table IV and V : (1) Regarding AP, AMR and F_1-score, the proposed algorithm outperforms other algorithms, which achieves an AP of 0.88, an AMR of 0.46, and a F_1 score of 0.9157 on the open field dataset. Compared to (Faster R-CNN + Transfer LFC), our proposed algorithm (Probabilistic Faster R-CNN + Transfer LFC + SRP) can improve the AP from 0.82 to 0.88 (an improvement of 7.3%) and reduce ARM from 0.59 to 0.46 (an improvement of 22.0%). (2) Compared to ACF, all deep learning algorithms are much better in terms of AP, AMR, and F_1-score, especially for the proposed algorithm. (3) Adding constraints of bounding boxes (e.g. height and width) can improve AP from 0.82 to 0.87 and reduce AMR from 0.59 to 0.51. However, integrating uncertainty from end to end

TABLE V
COMPARISON OF OBJECT LEVEL ON OPEN FIELD DATASET

Algorithm	Average Precision (AP)	Average Miss Rate (AMR)	Max F_1-score
ACF	0.49	0.89	0.5712
Faster R-CNN + Transfer LFC	0.82	0.59	0.8394
Ours (Faster R-CNN + Transfer LFC + BBC)	0.87	0.51	0.8796
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP without ICN)	**0.88**	**0.46**	**0.9157**

(Probabilistic Faster R-CNN + Transfer LFC + SRP) is a more efficient method which has better performance with regard to AP, AMR, and F_1-score compared with adding constraints (Faster R-CNN + Transfer LFC + BBC). (4) According to our experiment setting, the convolutional layers and first several fully connected layers of pre-trained neural network can be directly used for feature extraction without reinitialization. Transferring last fully connected layer provides better results than transferring all fully connected layers.

3) Image Level Performance: In contrast to the evaluation of object level, image level evaluation is to assess the performance of correctly recognising grasshopper(s) containing in an image. In this section, the proposed algorithm with an image classification network is also implemented and compared with the algorithms assessed in Section VII-B2. The comparative results are summarised in Table VI and VII based on precision, recall, specificity, accuracy and F_1-score.

Two key observations can be drawn from Table VI and VII: (1) The proposed algorithm with image classification network provides the best performance with regard to all image level evaluation performance. More specifically, our proposed algorithm can achieves an accuracy of 90.2%, a F_1 score of 0.9095, and a specificity of 80.6% in an open filed. Despite ACF yields a surprising good performance for the cage dataset, its performance significantly decreases for the open field. Compared to the ACF performance in the open field, our proposed algorithm improves 11.8% on accuracy, 30.4% on specificity, and 8.7% on F_1-score due to the powerful feature extraction property of deep CNNs [47]. (2) Compared to object detection purely using deep learning, ACF is more capable of correctly recognising grasshoppers contained in an image due to handcraft features of ACF performing well in object recognition (image level). However, it cannot locate the positions of grasshopper well within an image because conventional methods (e.g. slide windows) underperform in proposing fine region of interests (RoI) compared to region

proposal network which is a learning-based region proposal method. (3) It is evident that the ICN significantly improve the object detection performance at the image level. The reason is that pure deep convolutional neural networks for object detection focus on local information with neglecting the information from entire image, which lead to underperform our proposed method with ICN taking image level information into account through a image classification network.

TABLE VI

PERFORMANCE COMPARISON AT IMAGE LEVEL ON GLASS CAGE DATASET

Algorithm	Precision	Recall	Specificity	Accuracy	F_1-score
ACF	98.4%	100%	99.1%	99.4%	0.9919
R-CNN + Transfer AFC	61.4%	100%	63.1%	76.7%	0.7608
R-CNN + Transfer LFC	76.5%	100%	82.0%	88.7%	0.8669
Faster R-CNN + Transfer AFC	57.2%	100%	56.2%	72.4%	0.7277
Faster R-CNN + Transfer LFC	60.5%	100%	61.6%	75.8%	0.7539
Ours (Faster R-CNN + Transfer LFC + BBC)	60.5%	100%	61.8%	75.9%	0.7539
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP)	60.5%	100%	61.8%	75.9%	0.7539
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP + ICN)	**99.2%**	**100%**	**99.5%**	**99.7%**	**0.9960**

TABLE VII

PERFORMANCE COMPARISON AT IMAGE LEVEL ON OPEN FILED DATASET

Algorithm	Precision	Recall	Specificity	Accuracy	F_1-score
ACF	71.9%	100%	61.8%	80.7%	0.8365
Faster R-CNN + Transfer LFC	51.1%	100%	6.5%	52.7%	0.6764
Ours (Faster R-CNN + Transfer LFC + BBC)	51.1%	100%	6.5%	52.7%	0.6764
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP)	52.0%	100%	9.7%	54.4%	0.6842
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP + ICN)	**83.4%**	**100%**	**80.6%**	**90.2%**	**0.9095**

4) Object Level Performance Enhanced by Image Level Classification: An image classification network not only provides an accurate predictions of containing grasshopper(s) at image level classification, but also enhances the average precision while reducing average miss rate at object level. To quantitatively prove this argument, faster R-CNN, the proposed algorithm without an ICN and the proposed algorithm with an ICN are compared. The experimental results indicate that the proposed algorithm with an ICN has the best performance with AP of 0.91, AMR of 0.36, max F_1-score of 0.9263, which is displayed in Table VIII. Moreover, from Table VI and VII, we can see that the false positives can be reduced by using an ICN.

TABLE VIII

PERFORMANCE WITH USING IMAGE CLASSIFICATION NETWORK ON DATASETS

Algorithm	Average Precision (AP)	Average Miss Rate (AMR)	Max F_1-score
Glass Cage Grasshopper Dataset			
Faster R-CNN + Transfer LFC	0.77	0.58	0.7990
Ours (Faster R-CNN + Transfer LFC + BBC)	0.78	0.56	0.7979
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP)	**0.79**	**0.55**	**0.8059**
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP + ICN)	**0.81**	**0.49**	**0.8367**
Open Field Grasshopper Dataset			
Faster R-CNN + Transfer LFC	0.82	0.59	0.8394
Ours (Faster R-CNN + Transfer LFC + BBC)	0.87	0.51	0.8796
Ours (Probabilistic Faster R-CNN + SRP + Transfer LFC)	**0.88**	**0.46**	**0.9157**
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP + ICN)	**0.91**	**0.36**	**0.9263**

5) Computation time: In grasshopper detection, testing time is more important than training time particularly when the information is required for real-time variable rate treatment. We can see that our proposed algorithm (Probabilistic Faster R-CNN + Transfer LFC + SRP) outperforms others in testing time as shown in Table IX. This is because our proposed algorithm (Probabilistic Faster R-CNN + Transfer LFC + SRP) removes incorrect region proposals by considering the attributes (e.g. size and shape) and less region proposals are involved in executing Non-Maximum Suppression (NMS) compared to faster R-CNN. Such a design helps to reduce the computation load so our proposed algorithm (Probabilistic Faster R-CNN + Transfer LFC + SRP) outperforms others in testing time even though our algorithm looks more complicated. For training time, our proposed algorithm (Probabilistic Faster R-CNN + Transfer LFC + SRP) takes a little bit more time than faster R-CNN because it needs to train a GMM to model the attributes of region proposals. When our proposed algorithm (Probabilistic Faster R-CNN + Transfer LFC + SRP) is combined with image classification network (ICN), its performance (e.g. AP, AMR, and accuracy) is further improved with external computation load to train the ICN.

TABLE IX
COMPARISON OF COMPUTATION LOAD

Algorithm	Training Time (second)	Testing Time (second per image)
Faster R-CNN + Transfer LFC	1.3153×10^3	0.1141
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP)	1.3527×10^3	**0.1098**
Ours (Probabilistic Faster R-CNN + Transfer LFC + SRP + ICN)	1.4096×10^3	0.1610

VIII. CONCLUSIONS AND FUTURE WORK

This paper proposes a probabilistic faster R-CNN with stochastic region proposing, where the uncertainty of grasshopper(s) detection is passed from end to end. In the proposed algorithm, three networks are trained including: a probabilistic region proposal network, an image classification network, and an object detection network. The probabilistic region proposal network tunes the probability of proposing regions with the help of GMM, which improves the average precision whilst reducing the average miss rate as shown in experimental results of this paper. The image classification network returns the probability that grasshoppers are contained in an image, which performs well on image classification. The object detection network provides the confidence scores for each region proposal. The uncertainty of these three networks is fused by Bayesian probabilistic inference to derive final confidence scores for region proposals. Eventually, a proposed stochastic region proposing algorithm generates region proposals according to the final confidence scores, which is more robust compared to deriving region proposals by using a predetermined threshold.

The proposed algorithm is evaluated by using recently collected grasshopper datasets of different densities and under different light conditions. Both at object level and image level, the proposed algorithm achieves the best performance among all compared algorithms in an open filed. At object level, the proposed algorithm outperforms other algorithms in terms of average precision (0.91), average miss rate (0.36), and maximum F_1-score (0.9263). At image level, the proposed algorithm can extensively reduce the false positive rate for determining the existence of grasshoppers. The proposed method is also applicable to other remote sensing applications such as weed detection [29], crop yield estimation [30], animal/wildlife monitoring (e.g. sheep/cow detection and counting) [31], forest management [32] and other pest detection applications.

This paper mainly focuses on detecting grasshoppers with an RGB image and initially demonstrating its feasibility by using recently collected grasshopper datasets. There is still a room for

further development. For example, the proposed algorithm and architecture may be transferred to other remote sensing applications such as forest resource monitoring and plant disease management. The challenge is that these applications prefer to use a multispectral or hyperspectral camera rather than an RGB camera. Therefore, how to adapt the proposed algorithm on multispectral or hyperspectral image data will be the next work. As illustrated by experimental results in this paper, the proposed algorithm also can alleviate the problem caused by the lack of suitable loss function. For specific applications, it may be difficult to integrate prior information and human expert knowledge into a cost function. Moreover, it is challenging to tune parameters and to obtain global optimisation when embedding many terms into a cost function. Such a manner causes difficulty in balancing the terms of cost function, even leads to overfitting. Taking this into account, we design the proposed probabilitic model that allows us to embed prior knowledge, such as shape prior, objectness, global information (object recognition for image level), and object detection (object level) into the process pipeline so that they can integrate with deep learning features to improve the performance.

REFERENCES

[1] R. Kalamatianos, K. Kermanidis, I. Karydis, and M. Avlonitis, "Treating stochasticity of olive-fruit fly's outbreaks via machine learning algorithms," *Neurocomputing*, vol. 280, pp. 135–146, 2018.

[2] H. Calvo, M. A. Moreno-Armendáriz, and S. Godoy-Calderón, "A practical framework for automatic food products classification using computer vision and inductive characterization," *Neurocomputing*, vol. 175, pp. 911–923, 2016.

[3] X. Cheng, Y. Zhang, Y. Chen, Y. Wu, and Y. Yue, "Pest identification via deep residual learning in complex background," *Computers and Electronics in Agriculture*, vol. 141, pp. 351–356, 2017.

[4] L. Deng, Y. Wang, Z. Han, and R. Yu, "Research on insect pest image detection and recognition based on bio-inspired methods," *Biosystems Engineering*, vol. 169, pp. 139–148, 2018.

[5] T. Liu, W. Chen, W. Wu, C. Sun, W. Guo, and X. Zhu, "Detection of aphids in wheat fields using a computer vision technique," *Biosystems Engineering*, vol. 141, pp. 82–93, 2016.

[6] M. Ebrahimi, M. Khoshtaghaza, S. Minaei, and B. Jamshidi, "Vision-based pest detection based on svm classification method," *Computers and Electronics in Agriculture*, vol. 137, pp. 52–58, 2017.

[7] C. Wen, D. Wu, H. Hu, and W. Pan, "Pose estimation-dependent identification method for field moth images using deep learning architecture," *biosystems engineering*, vol. 136, pp. 117–128, 2015.

[8] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, 2019.

[9] W. Ding and G. Taylor, "Automatic moth detection from trap images for pest management," *Computers and Electronics in Agriculture*, vol. 123, pp. 17–28, 2016.

[10] D. Yi, J. Su, and W.-H. Chen, "Locust recognition and detection via aggregate channel features," 2019.

[11] G. Li and Y. Yu, "Contrast-oriented deep neural networks for salient object detection," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–14, 2018.

[12] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.

[13] Y. Miao, X. Tao, Y. Sun, Y. Li, and J. Lu, "Risk-based adaptive metric learning for nearest neighbour classification," *Neurocomputing*, vol. 156, pp. 33–41, 2015.

[14] D. Yi, J. Su, C. Liu, and W.-H. Chen, "New driver workload prediction using clustering-aided approaches," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 64–70, 2019.

[15] D. R. Nayak, R. Dash, and B. Majhi, "Brain mr image classification using two-dimensional discrete wavelet transform and adaboost with random forests," *Neurocomputing*, vol. 177, pp. 188–197, 2016.

[16] D. Yi, J. Su, C. Liu, and W.-H. Chen, "Personalized driver workload inference by learning from vehicle related measurements," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 159–168, 2017.

[17] D. Bouchaffra, "Mapping dynamic bayesian networks to α-shapes: Application to human faces identification across ages," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1229–1241, 2012.

[18] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 690–703, 2017.

[19] Y. Liu, D. Zhang, Q. Zhang, and J. Han, "Part-object relational visual saliency," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[20] Y. Liu, J. Han, Q. Zhang, and C. Shan, "Deep salient object detection with contextual information guidance," *IEEE Transactions on Image Processing*, vol. 29, pp. 360–374, 2019.

[21] K. Nogueira, O. A. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539–556, 2017.

[22] D. Tomè, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, and S. Tubaro, "Deep convolutional neural networks for pedestrian detection," *Signal Processing: Image Communication*, vol. 47, pp. 482–489, 2016.

[23] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 253–256.

[24] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[25] J. Wang, Z. Wang, X. Chen, and J. Qiu, "Synchronization criteria of delayed inertial neural networks with generally markovian jumping," *Neural Networks*, vol. 139, pp. 64–76, 2021.

[26] J.-G. Park and S. Jo, "Bayesian weight decay on bounded approximation for deep convolutional neural networks," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2866–2875, 2019.

[27] Y. Zhang, S. Pal, M. Coates, and D. Ustebay, "Bayesian graph convolutional neural networks for semi-

supervised classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5829–5836.

[28] W. Dong, X. Zhang, X. Zhang, H. Wu, M. Zhang, E. Ma, and J. Zhang, "Susceptibility and potential biochemical mechanism of oedaleus asiaticus to beta-cypermethrin and deltamethrin in the inner mongolia, china," *Pesticide biochemistry and physiology*, vol. 132, pp. 47–52, 2016.

[29] J. Yu, S. M. Sharpe, A. W. Schumann, and N. S. Boyd, "Deep learning for image-based weed detection in turfgrass," *European journal of agronomy*, vol. 104, pp. 78–84, 2019.

[30] E. Hamuda, B. Mc Ginley, M. Glavin, and E. Jones, "Improved image processing-based crop detection using kalman filtering and the hungarian algorithm," *Computers and Electronics in Agriculture*, vol. 148, pp. 37–44, 2018.

[31] F. Sarwar, A. Griffin, P. Periasamy, K. Portas, and J. Law, "Detecting and counting sheep with a convolutional neural network," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[32] A. A. d. Santos, J. Marcato Junior, M. S. Araújo, D. R. Di Martini, E. C. Tetila, H. L. Siqueira, C. Aoki, A. Eltner, E. T. Matsubara, H. Pistori *et al.*, "Assessment of cnn-based methods for individual tree detection on images captured by rgb cameras attached to uavs," *Sensors*, vol. 19, no. 16, p. 3595, 2019.

[33] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[34] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[35] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[37] G. Burel and D. Carel, "Detection and localization of faces on digital images," *Pattern Recognition Letters*, vol. 15, no. 10, pp. 963–967, 1994.

[38] B. Li, C. Liu, and W.-H. Chen, "An auxiliary particle filtering algorithm with inequality constraints," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4639–4646, 2016.

[39] D. Yi, J. Su, C. Liu, and W.-H. Chen, "Personalized driver workload inference by learning from vehicle related measurements," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 159–168, 2019.

[40] X. Zhao, Y. Hou, D. Song, and W. Li, "A confident information first principle for parameter reduction and model selection of boltzmann machines," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1608–1621, 2018.

[41] A. Penalver and F. Escolano, "Entropy-based incremental variational bayes learning of gaussian mixtures,"

IEEE transactions on neural networks and learning systems, vol. 23, no. 3, pp. 534–540, 2012.

[42] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[43] E. J. de Oliveira, I. Da Silva, J. L. R. Pereira, and S. Carneiro, "Transmission system expansion planning using a sigmoid function to handle integer investment variables," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1616–1621, 2005.

[44] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[46] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[47] P. N. Dawadi, D. J. Cook, and M. Schmitter-Edgecombe, "Automated cognitive health assessment using smart home monitoring of complex tasks," *IEEE transactions on systems, man, and cybernetics: systems*, vol. 43, no. 6, pp. 1302–1313, 2013.

Dewei Yi received a Ph.D degree from the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K. He is now a Lecturer with the Department of Computing Science at University of Aberdeen.

His current research interests include applied machine learning, personalized systems, hybrid intelligent systems, intelligent vehicles, vehicular network, and precise agriculture.

Jinya Su received his B.Sc. degree in Mathematics from Shandong University, China in 2011 and a Ph.D degree in the Department of Aeronautical and Automotive Engineering, Loughborough University, U.K. in 2016. From 2015, he was a research associate in the same institute. He joined the School of Computer Science and Electronic Engineering, University of Essex, as a lecturer in Computer Science and AI in 2019.

His research interests include autonomous systems and applied machine learning, and their real-world applications such as intelligent vehicle and precision agriculture.

Wen-Hua Chen received the M.Sc. and Ph.D. degrees from Northeast University, Shenyang, China, in 1989 and 1991, respectively. He is a professor with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K.

His current research interests include the development of advanced control, signal processing and decision making methods and their applications in aerospace engineering and precision agriculture.

He is a Chartered Engineering in the UK and fellow of Institute of Electrical and Electronics Engineers (IEEE), Institution of Engineering and Technology (IET) and Institution of Mechanical Engineers (IMechE).